

**LEONCIO JOSÉ DE ALMEIDA REIS**

**SISTEMA PARA PESQUISA DE PROTOCOLOS DE AVALIAÇÕES FÍSICAS**

Monografia apresentada como requisito parcial para conclusão do Curso de Licenciatura em Educação Física do Departamento de Educação Física, Setor de Ciências Biológicas, da Universidade Federal do Paraná.

**ORIENTADOR: Prof. Dr. WAGNER DE CAMPOS**

## **AGRADECIMENTO**

A meus pais e a minha irmã pelo carinho e principalmente pela paciência.

A todos os meus amigos e a todos aqueles que conviveram comigo dentro da UFPR, tornando esses quatro anos de vida acadêmica muito mais prazerosos. Com eles aprendi muito, não só na salas de aula e laboratórios mas principalmente nas festas, churrascos, viagens e outros momentos de alegria e descontração.

Aos professores do Departamento de Educação Física e de outros setores da federal, em especial a meu orientador Wagner de Campos, a professora da disciplina de monografia Simone Rechia, ao grande professor Fernando Mezzadri e ao futuro professor dessa universidade meu grande amigo Fernando Augusto Starepravo.

## **RESUMO**

Diversas áreas da educação física já desfrutam dos benefícios da informática, utilizando os computadores para as mais variadas finalidades. Muitos softwares já foram construídos exclusivamente para atender demandas nessa área, como programas que analisam os gases expirados por um indivíduo durante uma corrida ou que calculam os ângulos das articulações durante certo movimento. Um das áreas que necessitam do auxílio tecnológico do computador é a pesquisa de informações sobre protocolos de avaliações físicas. Observa-se que existe uma grande dificuldade, tanto para o profissional formado quanto o acadêmico, em se pesquisar qual protocolo utilizar. Os dados não estão centralizados e obter uma informação específica de algum protocolo é uma tarefa árdua e às vezes infundável, restando para esses pesquisadores aplicar somente aquilo que foi aprendido em sala de aula. Observando essa dificuldade e a possibilidade da utilização de recursos da computação para saná-la, foi criado um programa de computador capaz de armazenar informações sobre protocolos utilizados para avaliações físicas, facilitando a pesquisa de informações a respeito desse tema. O programa foi construído através da utilização da metodologia Orientada a Objetos e a linguagem de programação Java.

## SUMÁRIO

<b>1 APRESENTAÇÃO DO PROBLEMA .....</b>	<b>1</b>
1.1 JUSTIFICATIVA .....	2
1.2 OBJETIVO GERAL .....	3
1.2.1 objetivos Específicos .....	3
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>4</b>
2.1 A ANÁLISE ORIENTADA A OBJETOS .....	4
2.2 FASES DO DESENVOLVIMENTO DE UM SISTEMA EM UML .....	5
2.3 NOTAÇÃO EM UML E MODELO DE ELEMENTOS .....	7
2.3.1 Classes .....	8
2.3.2 Objetos .....	9
2.3.3 Relacionamentos .....	10
2.3.4 Diagramas .....	11
2.3.4.1 Diagrama de Casos de Uso .....	11
2.3.4.2 Diagrama de Classes .....	12
<b>3 FUNCIONAMENTO DO SISTEMA .....</b>	<b>13</b>
3.1 USUÁRIOS .....	13
3.2 EFETUAR LOGIN E LOGOUT .....	14
3.3 CADASTRO DE USUÁRIOS .....	14
3.4 CONTROLE DE ÁREAS .....	14
3.5 PROTOCOLOS .....	15
3.6 PESQUISA DE INFORMAÇÕES .....	16
<b>4 METODOLOGIA .....</b>	<b>17</b>
<b>REFERÊNCIAS .....</b>	<b>19</b>

## LISTA DE FIGURAS

Figura 1 – Representação de Class .....	9
Figura 2 – Representação de um objeto .....	9
Figura 3 – Representação de um relacionamento do tipo associação entre as classes Cliente e Conta Corrente .....	10
Figura 4 - Agregação entre duas classes .....	10
Figura 5 – Exemplo de um Diagrama de casos de uso .....	11
Figura 6 – Exemplo de um Diagrama de Classes .....	12

## **1 APRESENTAÇÃO DO PROBLEMA**

Desde o surgimento dos primeiros e gigantescos computadores na metade do século XX, a tecnologia para o desenvolvimento destes evoluiu de uma maneira extraordinária permitindo que aquelas antigas toneladas de metal viessem a ser substituídas por aparelhos eletrônicos cada vez menores e mais potentes. Esta evolução da informática levou por fim ao aparecimento do microcomputador, permitindo que a informática se popularizasse e atingisse quase todos os setores da vida econômica, social e política, modificando profundamente o modo de vida da sociedade (GADELHA, 2004). O computador, que antes custava milhões, passou a ser fabricado em série e consumido como um eletrodoméstico, sendo um instrumento presente e indispensável na classe média e acima. Aproximadamente 200 milhões de computadores de uso geral estão em utilização em todo o mundo, auxiliando pessoas no comércio, na indústria, no governo e em suas vidas pessoais, podendo esse número dobrar facilmente em alguns anos.

Muito mais que realizar cálculos aritméticos, os computadores são capazes de atuar em diversas áreas do conhecimento humano, simulando experiências e reações, auxiliando nas pesquisas laboratoriais, facilitando a troca e obtenção de informações, automatizando processos manuais, acelerando o processamento de informações, enfim, realizando em milésimo de segundos o que o cérebro humano só conseguiria após muito tempo e esforço. Tornam-se, portanto, ferramentas fundamentais para qualquer profissional, incluindo o atuante na área da Educação Física.

Diversas áreas da educação física já desfrutam dos benefícios da informática, utilizando os computadores para as mais variadas finalidades. Muitos softwares já foram construídos exclusivamente para atender demandas nessa área, como programas que analisam os gases expirados por um indivíduo durante uma corrida ou que calculam os ângulos das articulações durante certo movimento. Além desses, muitos outros nem tão específicos, são desenvolvidos para facilitar a coleta e armazenamento de informações, melhorando a atuação do profissional de educação física. No entanto, ainda há muito a ser criado, principalmente no que se refere à pesquisa de informações. A internet é uma ferramenta muito útil, mas a pesquisa virtual ainda

apresenta problemas sérios como a quantidade de lixo virtual e de informações desencontradas.

Dentro da Educação Física, uma das áreas que necessitam do auxílio tecnológico do computador é a pesquisa de informações sobre protocolos de avaliações físicas. Observa-se que existe uma grande dificuldade, tanto para o profissional formado quanto o acadêmico, em se pesquisar qual protocolo utilizar. Os dados não estão centralizados e obter uma informação específica de algum protocolo é uma tarefa árdua e às vezes infundável, restando para esses pesquisadores aplicar somente aquilo que foi aprendido em sala de aula. Observando essa dificuldade e a possibilidade da utilização de recursos da computação para saná-la, propõem-se a criação de um programa de computador capaz de armazenar informações sobre protocolos utilizados para avaliações físicas, facilitando a pesquisa de informações a respeito desse tema.

### 1.1 JUSTIFICATIVA

São inúmeras as situações em que o profissional de educação física necessita realizar periodicamente uma avaliação física de seus alunos (na academia, no clube, na escolinha de futebol, etc.), a fim de verificar a condição física destes e se os objetivos estabelecidos estão sendo alcançados. Isso mostra que a avaliação física é um importante instrumento de trabalho para o profissional de educação física, sendo indispensável para a boa atuação do mesmo. Entretanto, muitos profissionais deixam de avaliar, por exemplo, algumas capacidades físicas simplesmente por não conhecerem um protocolo que se enquadre nas condições da academia ou do clube. Outras vezes, utilizam um protocolo que é indicado para uma população diferente da avaliada. Este fato existe devido a grande dificuldade em se pesquisar qual protocolo utilizar. Os dados não estão centralizados e obter uma informação específica de algum protocolo é uma tarefa muito difícil. Para se ter idéia, uma pesquisa em um site de busca, o Google ([www.google.com.br](http://www.google.com.br)), utilizando as palavras-chaves “protocolos” e “flexibilidade”, encontrará cerca de 8 mil resultados. Substituindo a palavra “flexibilidade” por “força”, o número de referências encontradas supera 16 mil.

Para facilitar essa pesquisa de protocolos de avaliações físicas, pretende-se desenvolver um programa computadorizado capaz de armazenar informações referentes a estes protocolos. Através desse banco de dados eletrônico, o profissional ou acadêmico de educação física poderá realizar pesquisas para encontrar o protocolo de avaliação física mais adequado para determinada população, ou mais acessível para as recursos materiais que o avaliador tem a sua disposição.

## 1.2 OBJETIVO GERAL

Desenvolver um banco de dados informatizado que permita, através da utilização de palavras-chaves, a busca de protocolos utilizados para avaliações físicas.

### 1.2.1 Objetivos Específicos

- Permitir a alimentação dos dados, bem como a alteração / exclusão dos dados armazenados.
- Possibilitar um controle e cadastro de usuários.



## **2 FUNDAMENTAÇÃO TEÓRICA**

A fundamentação teórica desta monografia baseia no estudo da metodologia Orientada a Objetos, pois através dela foi feita toda análise, projeto e desenvolvimento do SPPAF - Sistema para Pesquisas de Protocolos de Avaliações Físicas. Cada capítulo pretende abordar determinadas características desta metodologia através de uma linguagem de fácil entendimento para um leigo em informática. Este estudo não foi muito aprofundado, termos técnicos e específicos do profissional de programação e análise de sistemas não foram mencionados. Trata-se, portanto de uma revisão teórica bem generalizada sobre a metodologia, buscando aproximar, sempre que possível, a forma com que foi utilizada na prática esta metodologia na construção do sistema.

### **2.1 A ANÁLISE ORIENTADA A OBJETOS**

A compreensão e interpretação correta de um problema são fundamentais para que ele seja resolvido. Esta afirmação é válida para a resolução de qualquer tipo de problema, seja na matemática, medicina, física, informática ou qualquer outra área de conhecimento. Dada esta importância, os projetistas de sistema sempre procuraram e ainda procuram técnicas aperfeiçoadas para a fase que antecede o desenvolvimento de sistemas: a compreensão perfeita do problema a ser resolvido (WINBLAD; EDWARDS; KING, 1993, p.194). Estas técnicas desenvolvidas que permitem e facilitam o entendimento correto do que deve ser feito são chamadas de metodologias de análise. Basicamente existem duas metodologias para análise de sistemas informatizados: a Análise Estruturada e a Análise Orientada a Objetos. Esta última técnica - que será utilizada para o desenvolvimento deste sistema – é mais recente que a outra e já é utilizada em inúmeros projetos com eficiência e qualidade comprovada.

A Análise Orientada a Objetos “ênfatiza a definição das características e do comportamento dentro de um sistema de objetos” (SHLAER & MELLOR, 1990). Através desta metodologia, as informações a serem processadas são pensadas como objetos manipuláveis, aproximando-os o máximo possível da realidade. Consiste basicamente em observar o que será utilizado no sistema de acordo com suas características no mundo real. Assim, por exemplo, um sistema com o objetivo de

armazenar dados antropométricos de crianças irá trabalhar com objetos “Alunos”. Cada objeto “Aluno” possuirá suas características (atributos) e suas ações comuns. Ao se aplicar a metodologia de análise Orientada a Objetos para o desenvolvimento do SPPAF, a primeira tarefa a ser realizada foi o levantamento das informações ou dados que seriam utilizados, interpretando-os como objetos: objeto Protocolo, Usuário, etc.

Quando surgiu, a Análise Orientada a Objetos não tinha padronização nenhuma, então Grady Booch, James Rumbaugh, e Ivar Jacobson decidiram criar uma padronização das metodologias de desenvolvimento de sistemas baseados na orientação a objetos. Criaram então a UML “Unified Modeling Language”. A UML nada mais é que uma maneira padrão de se construir sistemas orientados a objetos, fazendo com que os sistemas construídos com base na metodologia orientadas a objeto sejam construídos de maneira uniforme, facilitando o entendimento de outras pessoas que venham a estudar o software construído. Isto possibilita a estas, sem muitas dificuldades, fazer alguma modificação no sistema ou dar continuidade ao trabalho desenvolvido até então. Esta padronização incorpora as noções do desenvolvimento de software totalmente visual, baseando-se em diversos estilos de diagramas, cada um com suas próprias características, possibilitando um fácil entendimento do sistema, tanto no momento de sua criação quanto futuramente, quando analisado por outros (FURLAN, 1998).

## 2.2 FASES DO DESENVOLVIMENTO DE UM SISTEMA EM UML

O desenvolvimento de um sistema em UML divide-se em 5 fases: análise de requisitos, análise, design, implementação (programação) e testes. “Estas cinco fases não devem ser executadas na ordem descrita acima, mas simultaneamente de forma que problemas detectados numa certa fase modifiquem e melhorem as fases desenvolvidas anteriormente de forma que o resultado global gere um produto de alta qualidade e performance” (FURLAN, 1998).

- **Análise de requisito:** esta fase captura as intenções e necessidades dos usuários do sistema a ser desenvolvido através do uso de funções chamadas “casos de

uso”, que serão explicadas detalhadamente nos próximos capítulos. Nesta fase o analista do sistema deve entender exatamente o que o usuário deseja, permitindo que o projeto seja construído corretamente. Perguntas básicas como: para que serve? O que faz? Como será utilizado? Devem ser esclarecidas e facilmente entendidas através dos diagramas. O diagrama de casos de uso do sistema pode ser observado no Anexo 1. Esta fase envolve basicamente o(s) analista(s) do sistema e os usuários que o necessitam. Definiu-se o que o programa iria pesquisar, que tipos de protocolos seriam cadastrados e como seria acessado pelos usuários.

- **Análise:** a fase de análise está preocupada com as primeiras abstrações (classes e objetos) e mecanismos que estarão presentes no domínio do problema. É uma fase específica para as classes são modeladas e ligadas através de relacionamentos com outras classes, e são descritas no diagrama de classe (Anexo 2). A análise mostra o que deve ser feito para que o sistema atenda as necessidades estabelecidas na fase de análise de requisito.

Definiu-se, de uma maneira bem técnica, como seriam as estruturas dos protocolos armazenados no computador, e também, o tipo e tamanho dos campos: ex: campo nome com até 50 caracteres; campo população indicada com até 100 caracteres. Para isso foram pesquisados alguns protocolos de avaliações físicas para que servissem como modelos, permitindo que o programa seja construído de uma maneira que permita incluir outros inúmeros protocolos.

- **Design (projeto):** na fase de design, o resultado da análise é expandido em soluções técnicas. Novas classes serão adicionadas para prover uma infra-estrutura técnica: a interface do usuário e de periféricos, gerenciamento de banco de dados, comunicação com outros sistemas, dentre outros. as classes modeladas na são mescladas nessa nova infra-estrutura técnica, resultando no detalhamento das especificações para a fase de programação do sistema.

Nessa etapa, foram definidas e criadas a interfaces do usuário (telas). Elas fazem a conexão entre os dados fornecidos pelo usuário via teclado ou mouse e a execução de determinadas funções dentro do programa. Criou-se uma tela principal, na qual, através da utilização de menus, são executadas outras telas, cada uma sendo

responsável pela execução de determinada função do sistema: tela para inclusão/alteração e exclusão de protocolos e de áreas para esses protocolos; tela para pesquisa/consulta dos protocolos inclusos e tela para controle de usuários.

- **Programação:** na fase de programação, as classes provenientes do design são convertidas para o código da linguagem orientada a objetos escolhida. Dependendo da capacidade da linguagem usada, essa conversão pode ser uma tarefa fácil ou muito complicada. Nas fases anteriores, os modelos criados são o significado do entendimento e da estrutura do sistema, já a programação é uma fase separada e distinta onde os modelos criados são convertidos em código, ou seja, são convertidos para uma linguagem que o computador entenda.

- **Testes:** um sistema normalmente é rodado em testes de unidade, integração, e aceitação. os testes de unidade são para classes individuais ou grupos de classes e são geralmente testados pelo programador. Os testes de integração são aplicados já usando as classes e componentes integrados para se confirmar se as classes estão cooperando uma com as outras como especificado nos modelos. Os testes de aceitação são aqueles realizados pelo usuário final, que testam, observam e verificam se o sistema está funcionando como o desejado, ou seja, exatamente ao especificado nos primeiros diagramas de casos de uso. Os possíveis erros encontrados são corrigidos e então o sistema está concluído e pronto para ser utilizado.

## 2.3 NOTAÇÃO DE UML E MODELOS DE ELEMENTOS

Os sistemas orientados a objetos construídos de acordo com a UML devem seguir a notação definida pela UML, ou seja, as normas definidas para a representação gráfica (através dos diagramas) dos conceitos utilizados na construção do sistema e a relação entre esses conceitos. Os conceitos usados nos diagramas são chamados de modelos de elementos. Cada modelo de elemento possui características únicas e uma definição formal exata de seu significado. Isso permite a qualquer analista de sistema observar qualquer sistema construído sob a padronização da UML e identificar claramente todos aqueles elementos utilizados. Os modelos de elementos utilizados para a construção do SPPAF foram o seguinte:

### 2.3.1 Classes

Uma classe é a descrição de um tipo de objeto: aluno, paciente, protocolos, automóveis, etc, ou seja, é algo que reúne junto a si objetos que possuem características e comportamentos comuns. A classe Protocolo, criado para o desenvolvimento deste software, irá reunir centenas de objetos semelhantes que são os protocolos de avaliações físicas: protocolo para aferição de VO2max, dobras cutâneas, força, flexibilidade e outros. Todos os objetos são instâncias de classes, onde a classe descreve as propriedades e comportamentos daquele objeto. Deste modo, os atributos definidos para a classe Protocolo anteriormente citada, servem para todos os objetos reunidos nesta classe (todos os protocolos incluídos no programa), os atributos são: Nome (nome de identificação do protocolo), Descrição (descrição dos procedimentos de execução do protocolo), População Indicada (população indicada para utilização do protocolo), Material Utilizado (material necessário para realização do teste) e Referência (referências sobre o protocolo e do local que foi retirado).

Uma classe pode ser a descrição de um objeto em qualquer tipo de sistema – sistemas de informação, técnicos, integrados real-time, distribuídos, software, etc. Portanto, além das classes retiradas e construídas a partir do mundo real: eletrodomésticos, professores, etc., existem classes que representam entidades de software como arquivos, programas executáveis, janelas, barras de rolagem, botões, etc.

Identificar as classes de um sistema pode ser complicado, e deve ser feito por pessoas experientes no domínio do problema a que o software modelado se baseia. As classes devem ser retiradas do domínio do problema e serem nomeadas pelo que elas representam no sistema.

Segundo Furlan (1998), em UML as classes são representadas por um retângulo dividido em três compartimentos: o compartimento de nome, que conterá apenas o nome da classe modelada, o de atributos, que possuirá a relação de atributos que a classe possui em sua estrutura interna, e o compartimento de operações, que serão o métodos de manipulação de dados e de comunicação de uma classe com outras do sistema. Ao lado do atributo, define-se qual tipo de dados será armazenado naquele

atributo: string (aquele atributo irá armazenar informação na forma de texto), num (números inteiros), date (na forma de data), etc. (Figura 1). A sintaxe (maneira de dizer ao computador o que deve ser feito) usada em cada um destes compartimentos é independente de qualquer linguagem de programação. Uma vez definidas, as classes independem de qual linguagem será utilizada para sua construção, podendo ser C++, Java ou qualquer outra linguagem orientada a objetos.

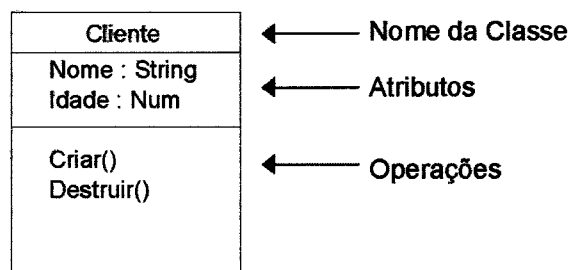


Figura 1 – Representação de Classe

### 2.3.2 Objetos

Um objeto é um elemento que podemos manipular, acompanhar seu comportamento, criar, destruir etc. Um objeto geralmente existe no mundo real. Pode ser uma parte de qualquer tipo de sistema, por exemplo, uma máquina, uma organização, ou negócio. Existem objetos que não encontramos no mundo real, mas que podem ser vistos de derivações de estudos da estrutura e comportamento de outros objetos do mundo real (COAD & YOURDON, 1992).

Em UML um objeto é mostrado como uma classe só que seu nome (do objeto) é sublinhado, e o nome do objeto pode ser mostrado opcionalmente precedido do nome da classe (FURLAN, 1998).

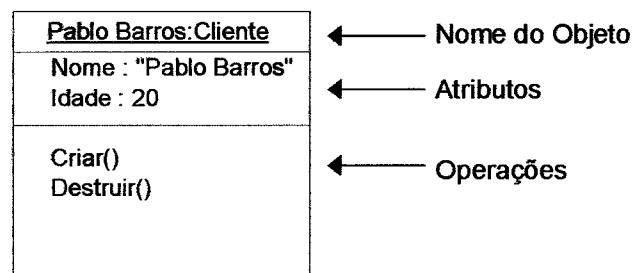


Figura 2 – Representação de um objeto

### 2.3.3 Relacionamentos

Os relacionamentos ligam as classes/objetos entre si criando relações lógicas entre estas entidades (BARROS,2001 apud ALMEIDA; DAROLT, 2001) . Existem vários tipos de relacionamentos, entretanto, somente os que serão utilizados para a construção do SPPAF serão descritos. São eles: associações e agregações.

#### Associações:

Segundo Furlan (1998) “uma associação representa que duas classes possuem uma ligação (link) entre elas, significando por exemplo que elas “conhecem uma a outra”, “estão conectadas com”, “para cada X existe um Y” e assim por diante.”

O tipo mais comum de associação é apenas uma conexão entre classes. É representada por uma linha sólida entre duas classes. A associação possui um nome (junto à linha que representa a associação), normalmente um verbo, mas substantivos também são permitidos.

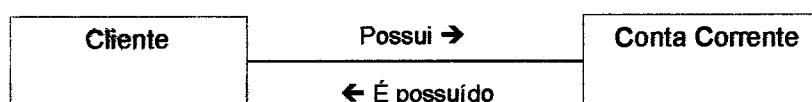


Figura 3 – Representação de um relacionamento do tipo associação entre as classes Cliente e Conta Corrente.

#### Agregação

A agregação é um caso particular da associação. A agregação indica que uma das classes do relacionamento é uma parte, ou está contida em outra classe, como mostra o exemplo na figura 4. As palavras chaves usadas para identificar uma agregação são: “consiste em”, “contém”, “é parte de”. No SPPAF um protocolo qualquer estará agregado a uma área (força, flexibilidade, % de gordura corporal, etc) e esta área possuirá outros inúmeros protocolos. Ex: A área força, estará associada a n protocolos que se referem a força.



Figura 4 – Agregação entre duas classes

### 2.3.4 DIAGRAMAS

São nove diagramas utilizados pela UML, cada um retratando determinada característica do sistema a ser programado. Além de facilitar a construção do sistema e a programação, são importantes ferramentas de documentação do sistema, permitindo que uma pessoa alheia a criação do sistema possa entender, após a observação dos diagramas, como o sistema funciona. Dois diagramas foram utilizados na criação do SPPAF: Diagrama de Classes e Diagrama de Casos de Uso.

#### 2.3.4.1 Diagrama de Casos de Uso

A modelagem de um diagrama casos de uso é uma técnica usada para descrever e definir os requisitos funcionais de um sistema. Eles são escritos em termos de atores externos, casos de usos e o sistema modelado. Os Atores representam o papel de uma entidade externa ao sistema como um usuário, um hardware, ou outro sistema que interage com o sistema modelado. Eles irão interagir, passando ou recebendo informações, com os Casos de Uso (ALMEIDA; DAROLT, 2001). Veja o exemplo na figura 5. O Diagrama de Casos de Uso do SPPAF pode ser consultado no Anexo 1.

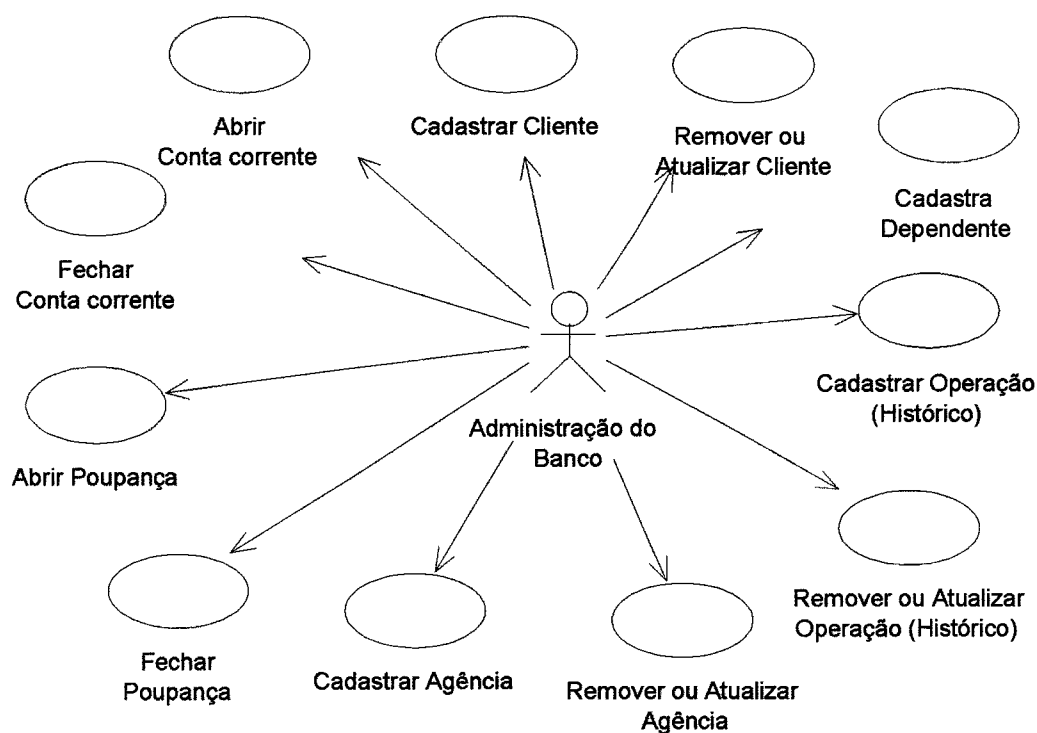


Figura 5 – Exemplo de um Diagrama de casos de uso



### 2.3.4.2 Diagrama de Classes

O diagrama de classes demonstra a estrutura das classes de um sistema e como elas se relacionam entre si. Todos os relacionamentos são mostrados no diagrama de classes juntamente com as suas estruturas internas, que são os atributos e operações. O Diagrama de Classes do SPPAF está no Anexo 2.

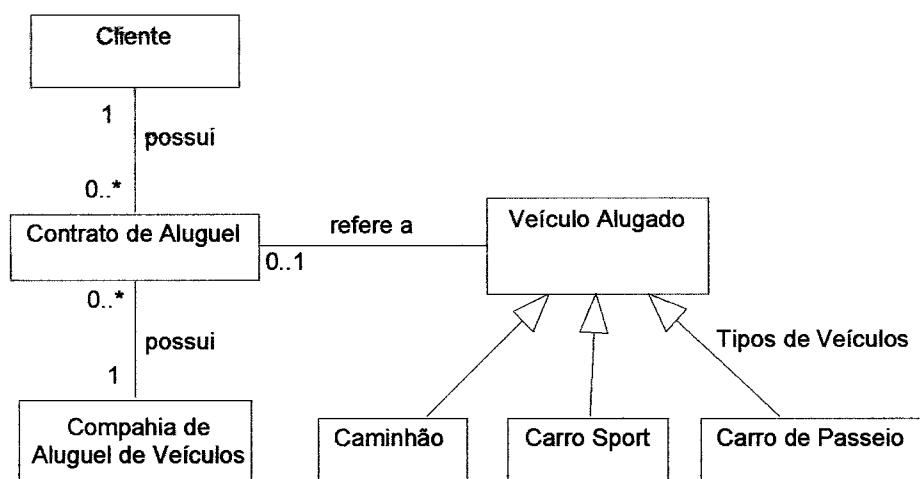


Figura 6 – Exemplo de um Diagrama de Classes

### 3 FUNCIONAMENTO DO SISTEMA

Este capítulo pretende mostrar toda a estrutura do sistema, para que serve e como deve ser executada cada função, desde o Login (entrada de um usuário no sistema) até a pesquisa de protocolos – função principal do programa.

#### 3.1 USUÁRIOS

**Usuário** é a definição para as pessoas que de fato utilizarão o sistema. Os usuários estão divididos em duas categorias distintas, de acordo com a função que irão exercer: *Administradores e Pesquisadores*. O Diagrama de Casos de Uso no Anexo 1 mostra perfeitamente a atuação desses dois tipos de usuários.

Os usuários do tipo *Administradores* são aqueles que irão gerenciar o SPPAF. Eles devem administrar toda a estrutura de protocolos existentes, ficando a cargo dos mesmos incluir um novo protocolo, excluir um protocolo existente ou alterar um protocolo já cadastrado. São, portanto, responsáveis pela alimentação dos dados e, consequentemente, pelo bom funcionamento do programa. Os *administradores* também devem controlar outros usuários que utilizarão o sistema, podendo incluir um novo usuário (do tipo *Administrador* ou *Pesquisador*) ou alterar / excluir um já existente.

Já os usuários do tipo *Pesquisador* serão aqueles que irão utilizar o SPPAF para realizarem suas pesquisas sobre os protocolos já cadastrados no sistema (pelo usuário *Administrador* responsável). Os *pesquisadores* não podem incluir novos protocolos ou alterar informação dos cadastrados, essas funções somente são permitidas aos *administradores*. Esta limitação garante a veracidade das informações e impede que algum usuário, por erro ou má fé, inclua algo errado ou altere qualquer tipo de informação. Os *pesquisadores* não têm nenhum tipo de controle sobre os outros usuários do sistema.

### 3.2 EFETUAR LOGIN E LOGOUT

Essa é a função inicial do programa, na qual o usuário se identifica através de um Login (apelido) e uma Senha. Tem por finalidade impedir a pesquisa de informação de pessoas não autorizadas e principalmente identificar se o usuário é do tipo Pesquisador ou Administrador, pois isso limitará o acesso em determinadas funções do programa, como a função Controle de Usuário que só é permitida aos Administradores. servindo para identificar quem estará utilizando o programa. A tela de Efetuar Login pode ser observada no Anexo 3.

Logout é a função oposta ao Login e deve ser executada após o usuário utilizar o sistema. Esta função fecha todas as funções existentes e aciona novamente a tela de Efetuar Login para que um novo usuário possa se identificar e utilizar o sistema.

### 3.3 CONTROLE DE USUÁRIOS

Esta é uma função exclusiva dos administradores. Quando é executada mostra uma tela de Cadastro de Usuários (veja Anexo 4) com uma lista contendo todos os usuários incluídos no sistema. Através do acionamento dos botões é possível: incluir um novo usuário no sistema, definindo sua permissão (Pesquisador ou Administrador); excluir um usuário selecionado na lista; alterar o nome, login, senha ou permissão do usuário selecionado.

### 3.4 CONTROLE DE ÁREAS

Também é uma função exclusiva dos administradores. Quando é executada mostra uma tela de Cadastro de Áreas (veja Anexo 6) com uma lista contendo todas as áreas incluídas no sistema: Flexibilidade, Força, Vo2Max, etc.. Através do acionamento dos botões é possível: incluir uma nova área no sistema, para que posteriormente sejam adicionadas novos protocolos para esta área; excluir uma área selecionada na lista (isto irá excluir também todos os protocolos pertencentes a esta área); alterar o nome da área selecionada.

### 3.5 PROTOCOLOS

Os protocolos a serem incluídos no sistema terão os seguintes campos (forma como será armazenado no programa):

- **Nome:** nome do protocolo. Ex: Guedes, 1997.
- **População Indicada:** população indicada para aquele protocolo. Ex: Crianças e adolescentes de 7 a 18 anos.
- **Material Utilizado:** material necessário para a realização dos testes e obtenção dos resultados. Ex: compasso de dobras cutâneas.
- **Descrição:** descrição completa do procedimento que deve ser executado, das fórmulas necessárias, etc.
- **Referências:** referências das quais foram obtidas as informações sobre este protocolo. Ex: GUEDES, D. P. Crescimento Composição Corporal e Desempenho Motor de Crianças e Adolescentes. São Paulo: CLR Balieiro, 1997.

A inclusão de novos protocolos é uma função permitida somente para os usuários do tipo Administrador. Quando este pretende incluir um novo protocolo, deve pesquisar todas as informações a respeito do mesmo e completar os campos citados anteriormente com as respectivas informações. Também é possível ao Administrador excluir um protocolo já incluso ou alterar qualquer informação do protocolo.

Observe no Anexo 7 a tela que tem a finalidade de controlar a inclusão, exclusão e alteração dos protocolos. Esta tela contém duas listas: a primeira refere-se às áreas incluídas no programa e a segunda aos protocolos. Quando uma área é selecionada com o clique do mouse aparecerá na Lista de Protocolos todos os protocolos que estão incluídos naquela área. A partir desta última lista o Administrador pode selecionar um protocolo e, através do acionamento dos botões, excluí-lo ou alterar suas informações. Quando um protocolo é selecionado todas as informações referentes a este são mostrados nos campos localizados abaixo das listas. O botão Novo permite ao Administrador inserir um novo protocolo no programa pertencente a área selecionada na Lista de Áreas.

### 3.6 PESQUISA DE INFORMAÇÕES

Essa é uma das principais funções do programa pois é através dela que qualquer usuário (Administrador ou Pesquisador) poderá pesquisar informações sobre os protocolos cadastrados no sistema. Quando é executada abre uma janela (veja Anexo 8) contendo 5 campos de texto e duas listas. Nos campos o pesquisador deverá digitar a palavra que ele deseja encontrar em algum protocolo e em seguida selecionar a área específica em que ele deseja pesquisar. Os botões ao lado da caixa devem estar “tickados” indicando que o computador pesquisará aquele campo em seu banco de dados. Finalmente basta clicar no botão Pesquisar e na lista postada à direita aparecerá todos os protocolos que possuam aquele tipo de informação. Podem ser digitadas mais de uma palavra em cada campo como por exemplo: “idosos crianças” e computador ignora maiúsculas e minúsculas. O texto digitado não é tratado como uma palavra então se for digitado “dos” todas as palavras que contenham essas letras serão encontrados como “idosos”, “encontrados”, “dosagem”. Portanto se for digitado “adult” serão encontrados “adultos”, “adultas”.

## 4 METODOLOGIA

Assim como num trabalho de pesquisa, o desenvolvimento de *softwares* exige o suporte teórico de uma metodologia. Após ter definido o tema e coletado as informações necessárias para o desenvolvimento do sistema, escolher a metodologia a ser utilizada e aplicá-la corretamente é fundamental para que o software seja construído rapidamente, com um mínimo de erros possíveis e atenda a todas as necessidades a que se propõe. É através do método escolhido que será definido como o *software* será analisado, projetado e implementado (construído).

Optou-se pela metodologia Orientada a Objetos. Através desta metodologia, as informações a serem processadas são pensadas como objetos manipuláveis com suas características (atributos) e ações próprias. Assim, por exemplo, um sistema com o objetivo de armazenar dados antropométricos de crianças irá trabalhar com objetos “Alunos”. Esta técnica permite aproximar muito os dados à realidade, facilitando o entendimento dos mesmos. Quando construídos corretamente, sistemas orientados a objetos são flexíveis a mudanças, oportunizando a criação e implementação de componentes totalmente reutilizáveis. Isto é muito importante já que o sistema a ser desenvolvido é bem delimitado, entretanto, se for bem analisado e projetado, pode vir a ser ampliado ou modificado facilmente. As técnicas praticadas por uma análise e implementação orientadas a objetos evoluíram muito desde seu surgimento, permitindo a elaboração de sistemas cada vez mais complexos. Em contrapartida, métodos de análises convencionais “tornam-se cada vez mais difíceis de serem pensados à medida que os sistemas ficam mais complexos, até que, por fim, limitam a complexidade que podemos manipular.” (MARTIN & ODELL, 1995, p. 4).

Após a definição da metodologia, outra importante decisão refere-se à escolha da linguagem de programação a ser utilizada. Uma linguagem de programação é um conjunto de termos, palavras e símbolos que irão instruir o processador a executar determinada função. Existem dezenas de linguagens de programação. Estas linguagens surgiram e vão surgindo com a evolução dos computadores. Algumas delas são extremamente complexas, utilizadas na criação de grandes sistemas, e outras tão práticas que são mais utilizadas como ferramentas de ensino. A linguagem de

programação a ser utilizada é Java. Esta linguagem surgiu em 1991 e sua utilização cresceu muito com a explosão da Internet, dada a grande qualidade dessa linguagem para a criação de aplicativos on-line e de seu potencial para a criação de páginas WEB. (DEITEL & DEITEL, 2001, p. 64). Java é uma linguagem totalmente orientada a objetos, permitindo uma perfeita aplicação da análise e do planejamento realizados com base na metodologia orientada a objetos.

## REFERÊNCIAS

ALMEIDA, A.; DAROLT, R. **Pesquisa e Desenvolvimento em UML**. Monografia. Araraguá. 2001. Disponível em: <<http://www.mednet.com.br/ivan/www/apostilas/UML2.pdf>>. Acesso em: 09 de agosto de 2004. UNISUL – Universidade do Sul de Santa Catarina.

COAD, P. & YOURDON, E. **Análise Baseada em Objetos**. 2. ed. Rio de Janeiro. Editora Campus Ltda, 1992.

DEITEL, H. M. & DEITEL, P. J. **JAVA Como Programar**. Porto Alegre. Bookman, 2001.

FUNEL, M. **Java e banco de dados: teoria e prática**. Rio de Janeiro. Brasport, 1999.

FURLAN, J. D. **Modelagem de Objetos através da UML – the Unified Modeling Language**. São Paulo. Makron Books, 1998.

GADELHA, J. **A Evolução dos Computadores**. [s.l.];[s.d.]. Disponível em: <<http://www.ic.uff.br/~aconci/evolucao.html>>. Acesso em: 17 de maio de 2004.

MARTIN, J. & ODELL, J. J. **Análise e Projeto Orientados a Objeto**. São Paulo. Makron Books, 1995.

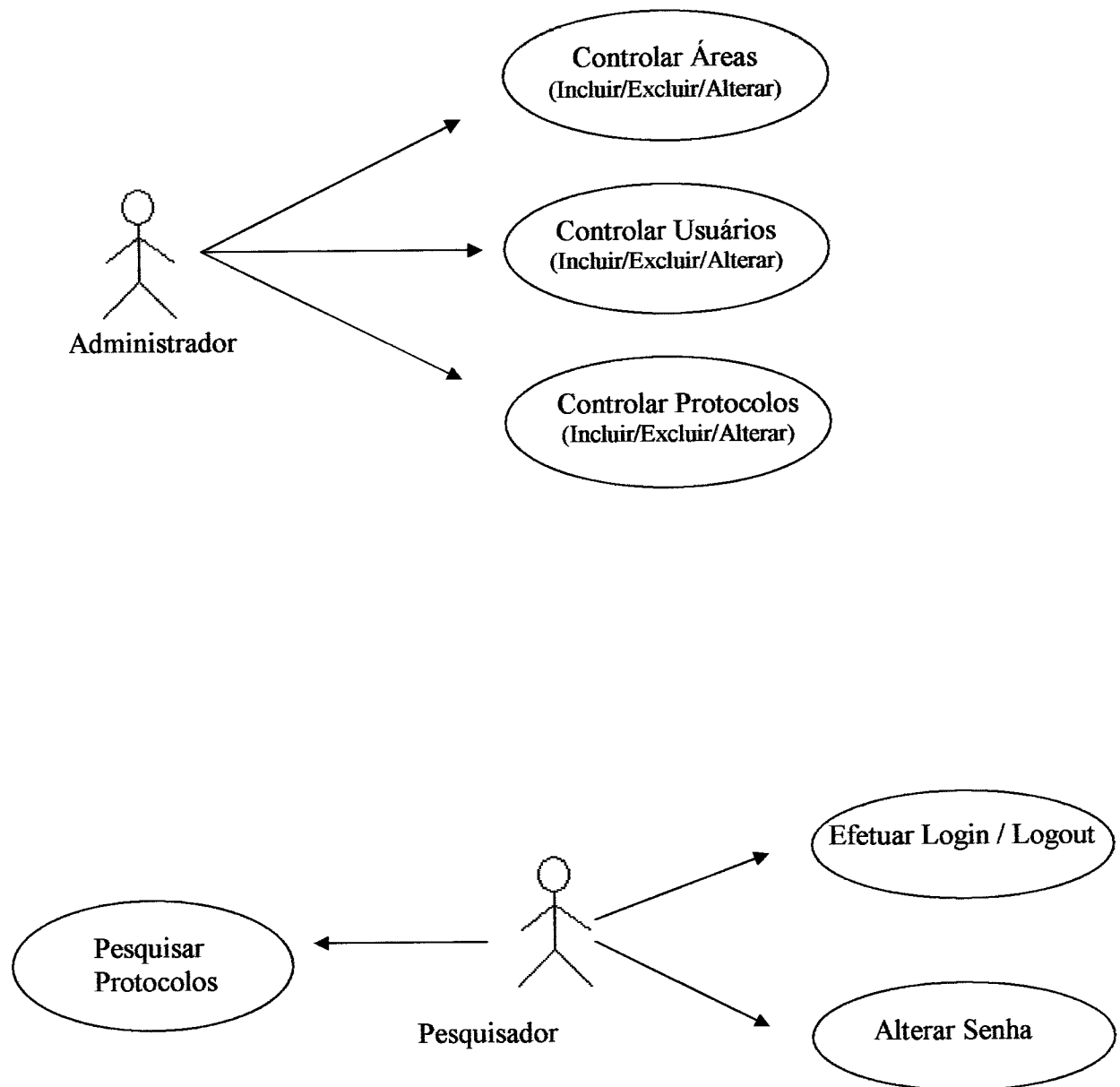
SHLAER, S. & MELLOR, S. **Análise de Sistemas Orientada para Objetos**. São Paulo. McGraw-Hill, 1990.

WINBLAD, A. L.; EDWARDS, S. D.; KING, D. R. **Software Orientado ao Objeto**. São Paulo. Makron Books, 1993. Trad. Denise de Souza Boccia.

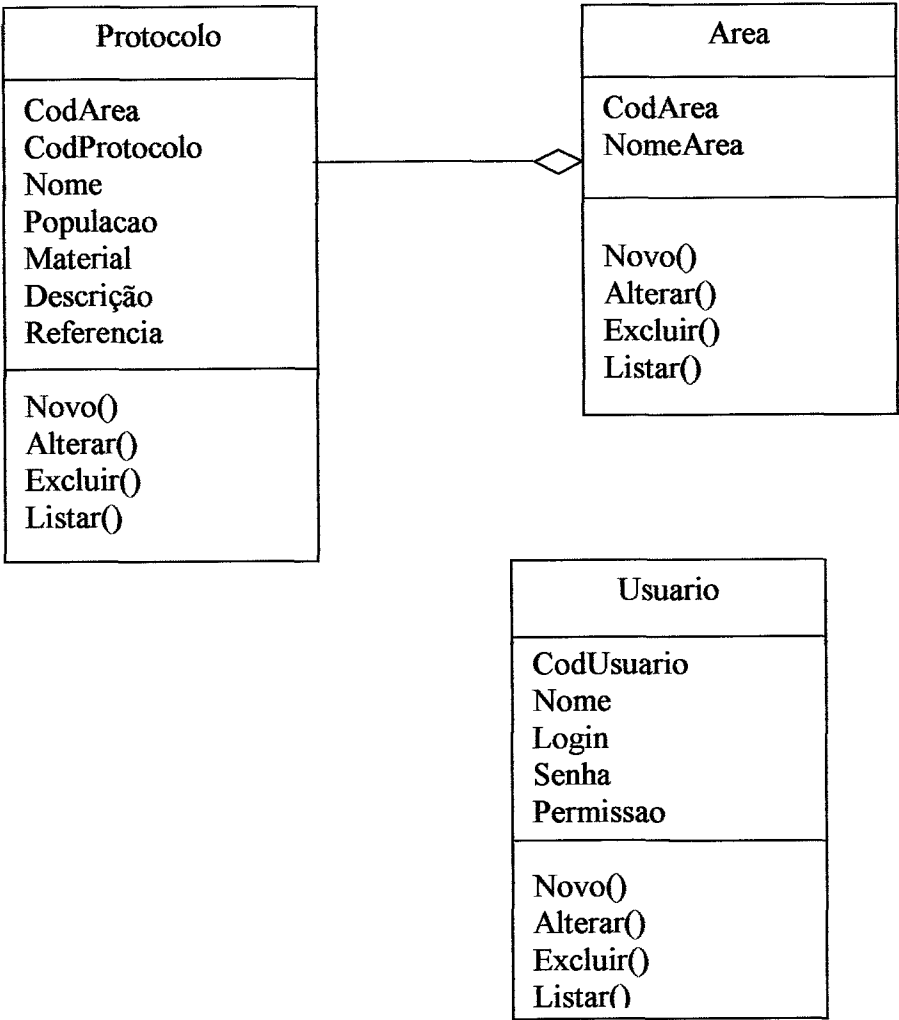


## **ANEXOS**

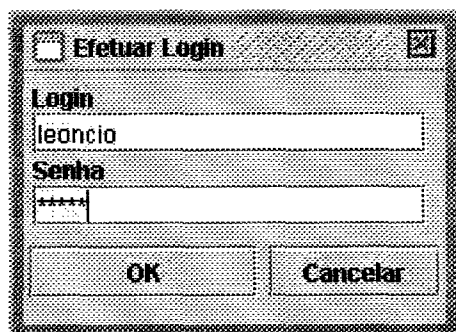
## ANEXO 1 – Diagrama de Casos de Uso



**ANEXO 2 – Diagrama de Classes**

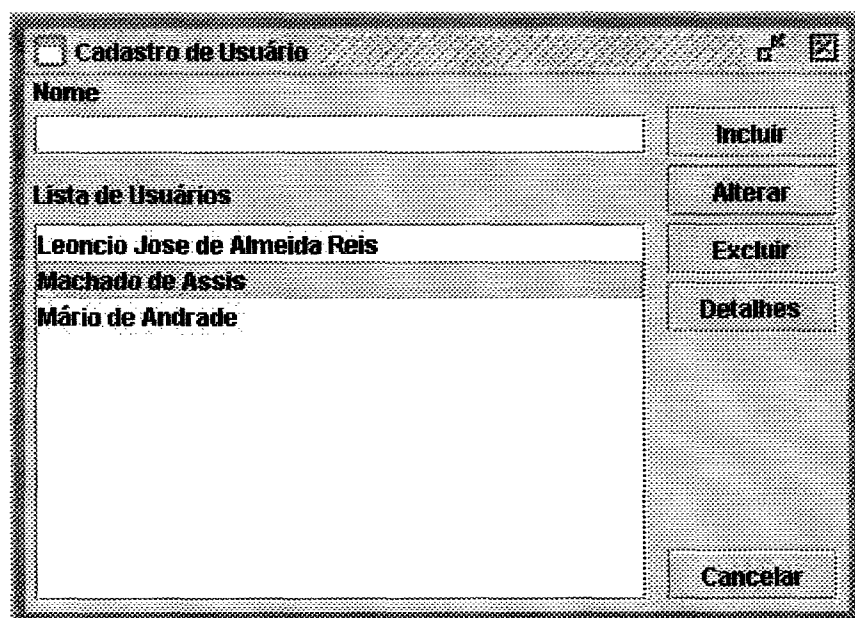


### ANEXO 3 – Imagem da tela Efetuar Login



A screenshot of a Windows-style dialog box titled "Efetuar Login". The dialog has a title bar with a minimize button, a maximize button, and a close button. Inside, there is a label "Login" above a text input field containing the text "leancio". Below this is a label "Senha" above a password input field filled with asterisks. At the bottom, there are two buttons: "OK" and "Cancelar".

### ANEXO 4 – Imagem da tela Cadastro de Usuários



A screenshot of a Windows-style dialog box titled "Cadastro de Usuário". The dialog has a title bar with a minimize button, a maximize button, and a close button. Inside, there is a label "Nome" above a text input field. Below this is a label "Lista de Usuários" above a list box containing three names: "Leoncio Jose de Almeida Reis", "Machado de Assis", and "Mário de Andrade". To the right of the list box are four buttons: "Incluir", "Alterar", "Excluir", and "Detalhes". At the bottom right, there is a "Cancelar" button.

## ANEXO 6 – Imagem da tela Cadastro de Áreas

☐ Cadastro de Áreas

Nome

Incluir

Lista de Áreas

% Gordura

Flexibilidade

Força

VO2 Max

Alterar

Excluir

Fechar

## ANEXO 7 – Imagem da tela Controle de Protocolos

☐ Protocolos

**Áreas:**

% Gordura  
Flexibilidade  
Força  
VO2 Max

**Lista de Protocolos:**

Guedes, 1997  
Jackson e Pollock, 1978; Siri, 1961  
Wetman e col., 1988

**Novo**  
**Excluir**  
**Salvar**  
**Fechar**

**Nome:** Guedes, 1997

**População Indicada:** Crianças e Adolescentes de 7 a 18 anos

**Material Utilizado:** Plicômetro

**Descrição:** Quando o somatório das espessuras das dobras cutâneas tricipital e subescapular for dividido pelo número de dobras, obtém-se o índice de gordura corporal. Este índice é comparado com os valores de referência para a população estudada.

Rapazes brancos:  
Pré-pubere % Gord =  $1,21(S) - 0,008(S)e^{2-1,7}$   
Púbere % Gord =  $1,21(S) - 0,008(S)e^{2-3,4}$   
Pós-Pubere % Gord =  $1,21(S) - 0,008(S)e^{2-5,5}$

Rapazes negros:  
Pré-pubere % Gord =  $1,21(S) - 0,008(S)e^{2-3,5}$   
Púbere % Gord =  $1,21(S) - 0,008(S)e^{2-5,2}$   
Pós-Pubere % Gord =  $1,21(S) - 0,008(S)e^{2-6,8}$

**Referências:** GUEDES, D. P. Crescimento Composição Corporal e Desempenho Motor de C

## ANEXO 8 – Imagem da tela de pesquisa

The image shows a software window titled "Pesquisa de Protocolos". It contains several input fields and checkboxes on the left, a search button in the center, and two list boxes on the right. The input fields are for "Nome:", "População indicada:", "Material utilizado:", "Descrição:" (checked), and "Referências:". The "Descrição:" field contains the text "Ariceps". Below these fields is a section labeled "Área:" containing a list of categories: "% Gorduras", "Flexibilidade", "Força", and "VO2 Max". The search button is labeled "Pesquisar". To the right of the search button is a list box titled "Protocolos encontrados" containing three entries: "Faulkner (%G em nadadores americanos)", "Yuhasz (%Gordura 6 dobras)", and "Pollock e Col (% gordura)". A "Fechar" button is located at the bottom right of the window.

☐ Pesquisa de Protocolos

Nome:

População indicada:

Material utilizado:

☒ Descrição: Ariceps

☐ Referências:

**Área:**

- % Gorduras
- Flexibilidade
- Força
- VO2 Max

**Protocolos encontrados**

- Faulkner (%G em nadadores americanos)
- Yuhasz (%Gordura 6 dobras)
- Pollock e Col (% gordura)